

We claim:

5

10

15

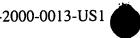
20

- A method for efficiently handling high contention locking in a multiprocessor computer system, comprising:
 organizing at least some of the processors into a hierarchy;
 providing a lock selected from the group consisting of: an interruptible lock, and a lock which waits using only local memory; and processing the lock responsive to the hierarchy.
- 2. The method of claim 1, wherein the processing step conditionally acquires the lock.
 - 3. The method of claim 2, wherein the processing step returns a failure to grant the lock if the lock is not immediately available.
 - 4. The method of claim 1, wherein the processing step unconditionally acquires the lock.
- 5. The method of claim 4, wherein the processing step spins on the lock until the lock is available.
 - 6. The method of claim 4, further comprising allowing system interrupts while spinning on the lock
- 7. The method of claim 1, wherein the processing step unconditionally releases the lock.
 - 8. The method of claim 1, wherein the processing step the processors spin on private memory.
 - 9. The method of claim 1, wherein the hierarchy includes a data structure having a bit mask indicating which processors of a group are waiting for the lock.
- The method of claim 1, wherein the hierarchy includes a data structure having a bit mask indicating which groups of processors have processors waiting for the lock.

5

10

15



- The method of claim 1, further comprising maintaining a release flag for a group 11. of processors to prevent races between acquisition and release of the lock.
- The method of claim 1, further comprising maintaining a handoff flag for a group 12. of processors to grant the lock to a processor requesting an unconditional lock from a processor requesting a conditional lock.
- A computer system comprising: 13. multiple processors;

a lock selected from the group consisting of: an interruptible lock, and a lock which waits using only local memory; and;

a hierarchical representation of processor organization; and a lock primitive for processing the lock responsive to the hierarchy.

- The computer system of claim 13, wherein said primitive further comprises a 14. conditional lock acquisition primitive.
- The computer system of claim 14, wherein said conditional lock acquisition 15. further indicates a lock failure if said lock is not immediately available.
- The computer system of claim 13, wherein said primitive further comprises an 16. unconditional lock acquisition primitive.
- The computer system of claim 16, wherein said processor may enter a spin stage 17. of said lock is not immediately available.
- The computer system of claim 16, wherein said lock may be subject to a system 20 18. interrupt during a spin stage.
 - The computer system of claim 13, wherein said primitive further comprises a 19. primitive for an unconditional release of said lock.
- The computer system of claim 13, wherein said primitive further comprises a 20. release flag to prevent races between acquisition and release of the lock. 25
 - The computer system of claim 13, wherein said primitive further comprises a 21. handoff flag to grant a lock to a processor requesting an unconditional lock from a processor requesting a conditional lock.

5



a computer-readable signal bearing medium;

means in the medium for hierarchically organizing at least some of the processors of a computer system;

means in the medium for providing a lock selected from the group consisting of: an interruptible lock, and a lock which waits using only local memory; and means in the medium for processing the lock responsive to the hierarchy.

- 23. The article of claim 22, wherein the medium is a recordable data storage medium.
- 24. The article of claim 22, wherein the medium is a modulated carrier signal.
- The article of claim 22, wherein the means is a conditional lock acquisition primitive.
 - 26. The article of claim 25, wherein a lock failure is indicated if the lock is not immediately available.
- The article of claim 22, wherein the means is an unconditional lock acquisition primitive.
 - 28. The article of claim 28, wherein a .spin stage is entered by a processor if the lock is not immediately available.
- 29. The article of claim 22, wherein the means is an unconditional lock release primitive.
 - 30. The article of claim 22, wherein said means is a release flag responsive to races between acquisition and release of a lock.
- The article of claim 22, wherein said means is a handoff flag responsive to a processor requesting an unconditional lock from a processor requesting a conditional lock.